Corso di Linguaggi di Programmazione — Parziale di fine modulo
Prova scritta del 24 Maggio 2021

Tempo a disposizione: 2 ore e 30 minuti.

1. Descrivere le regole di semantica operazionale strutturata per l'espressione booleana b_0 **xor** b_1 secondo la disciplina di valutazione esterna-destra (ED). Ricordo che b_0 **xor** b_1 vale vero se solo uno dei suoi argomenti vale vero. Per espressioni di questo tipo, valutarle con le regole ED o con le regole IS (interna-sinistra) può fare differenza? Argomentare la risposta.

2. Costruire una grammatica G che generi il linguaggio $L = \{a^{n+1}b^n a^m b^{m+k} \mid n, k \geq 0, m \geq 1\}$.

3. Classificare il linguaggio L del punto precedente, ovvero dire se L è regolare, oppure libero ma non regolare, oppure non libero, giustificando adeguatamente la risposta.

4. Si consideri l'espressione regolare $b^*(a|\epsilon)b^*$. Si costruisca l'automa NFA M associato, secondo la costruzione vista a lezione. Si trasformi l'NFA M nell'equivalente DFA M', secondo la costruzione per sottoinsiemi vista a lezione.

5. Preso il DFA M' calcolato al punto precedente, si verifichi se è minimo; se non lo fosse, lo si minimizzi per ottenere un DFA M''; poi si ricavi da M'' la grammatica regolare associata, seguendo la costruzione vista a lezione; quindi si semplifichi la grammatica ottenuta, eliminando i simboli inutili; infine, si ricavi dalla grammatica l'espressione regolare associata.

6. Dati i linguaggi L_1 ed L_2, il primo libero e il secondo regolare, a quale classe appartiene il linguaggio differenza $L_1 - L_2 = \{w \mid w \in L_1 \wedge w \notin L_2\}$? Motivare la risposta.

7. Mostrare che $L = \{a^{n+1}b^n c^m \mid n, m \geq 0\}$ è libero deterministico, costruendo un opportuno DPDA che riconosca $L\$$ per pila vuota. È possibile costruire un DPDA che riconosca L per pila vuota?

8. Si consideri la seguente grammatica G con simbolo iniziale S:

$$\begin{array}{lcl} S & \to & AB \mid BC \\ A & \to & \epsilon \mid \mathtt{b}D \\ B & \to & \mathtt{a} \mid \mathtt{b}CB \\ C & \to & \epsilon \mid C\mathtt{d} \\ D & \to & \mathtt{c} \mid \mathtt{d}SD \end{array}$$

   (*i*) Si calcolino i First e i Follow per tutti i nonterminali. (*ii*) La grammatica G è di classe LL(1)? (*iii*) Si rimuova la ricorsione sinistra immediata per C. (*iv*) Si rimuovano le produzioni epsilon per ottenere una grammatica G' senza produzioni epsilon, che sia equivalente a G.

9. Si consideri la grammatica G con simbolo iniziale S:

$$\begin{array}{lcl} S & \to & \mathtt{c}AS \mid \epsilon \\ A & \to & \mathtt{a} \mid \mathtt{a}A \end{array}$$

   (*i*) Determinare una espressione regolare che denoti il linguaggio generato $L(G)$. (*ii*) Verificare che G non è di classe LL(1). (*iii*) Manipolare la grammatica per ottenerne una equivalente G' di classe LL(1). (*iv*) Costruire la tabella di parsing LL(1) per G'. (*v*) Mostrare il funzionamento del parser LL(1) su input *caa*.

10. Si consideri la grammatica G del punto precedente. (*i*) Costruire l'automa canonico LR(0). (*ii*) Costruire la tabella di parsing SLR(1) e verificare se ci sono conflitti. (*iii*) Mostrare il funzionamento del parser SLR(1) per l'input *caa*.

11. Discutere la seguente affermazione: se L_1 è regolare e $L_1 \cap L_2$ è regolare, allora L_2 è sicuramente regolare.

1) Regole ED (Esterna Destra) per $\underline{xor}$

$$\frac{\langle b_1, \sigma\rangle \to \langle b_1', \sigma'\rangle}{\langle b_0 \text{ xor } b_1, \sigma\rangle \to \langle b_0 \text{ xor } b_1', \sigma'\rangle}$$

$$\frac{-}{\langle b_0 \text{ xor } tt, \sigma\rangle \to \langle \sim b_0, \sigma\rangle}$$

$$\frac{-}{\langle b_0 \text{ xor } ff, \sigma\rangle \to \langle b_0, \sigma\rangle}$$

$$\frac{\langle b_0, \sigma\rangle \to \langle b_0', \sigma'\rangle}{\langle \sim b_0, \sigma\rangle \to \langle \sim b_0', \sigma'\rangle}$$

$$\frac{-}{\langle \sim t, \sigma\rangle \to \langle t', \sigma\rangle}$$

t	t'
tt	ff
ff	tt

Per le espressioni b_0 xor b_1 vale che

$eval_{IS} = eval_{ED}$! Infatti, anche con la ED devo sempre valutare sia b_1 che b_0. L'ordine è diverso, ma se alla fine si ottiene un risultato, questo è lo stesso per entrambe! Esempio:

$\langle 3=2 \text{ xor } tt, \sigma\rangle \to_{ED} \langle \sim(3=2), \sigma\rangle \to_{ED} \langle \sim ff, \sigma\rangle \to_{ED} \langle tt, \sigma\rangle$

$\langle 3=2 \text{ xor } tt, \sigma\rangle \to_{IS} \langle ff \text{ xor } tt\rangle \to_{IS} \langle tt, \sigma\rangle$

2) $L = \{ a^{n+1} b^n a^m b^{m+k} \mid n, k \geq 0,\ m \geq 1 \}$

$S \to AB$

$A \to a \mid aAb$

$B \to CD$

$C \to ab \mid aCb$

$D \to \varepsilon \mid bD$

$L(D) = b^*$ $\quad L(C) = \{ a^m b^m \mid m \geq 1 \}$

$L(B) = \{ a^m b^{m+k} \mid m \geq 1, k \geq 0 \}$

$L(A) = \{ a^{n+1} b^n \mid n \geq 0 \}$

$L(S) = L$

---

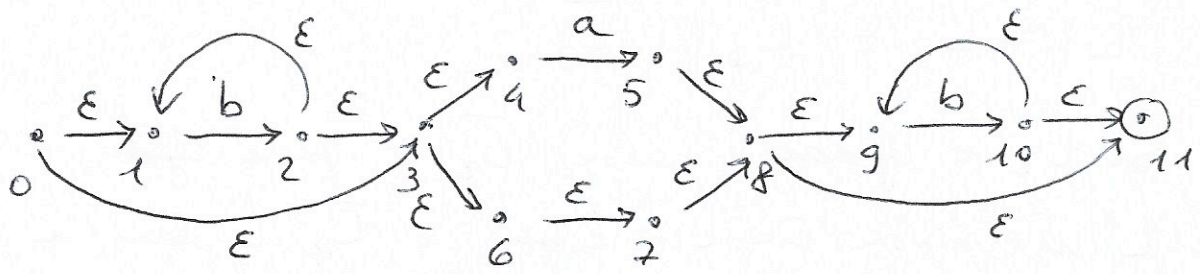
3) L del punto precedente è libero perché generato da una grammatica libera. L non è regolare e lo vediamo usando il pumping lemma al contrario.

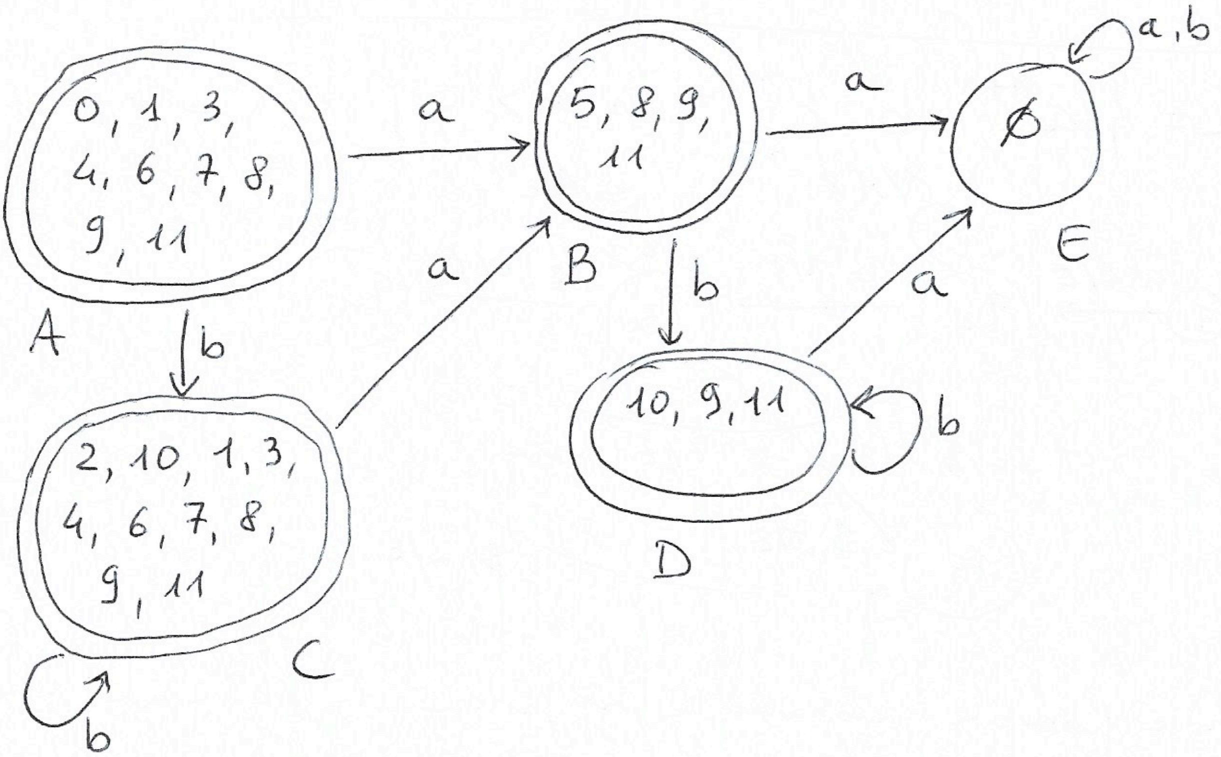
- Fissiamo $N > 0$ generico $\quad (\forall N > 0)$
- Scegliamo $z = a^{N+1} b^N a b \quad (\exists z \in L, |z| \geq N)$
- Per ogni u, v, w tali che
  - $z = uvw$
  - $|uv| \leq N$
  - $|v| \geq 1$

  deve essere $v = a^i$ con $i \geq 1$ (v dentro al blocco a^{N+1})
- $\exists k = 2$ tale che $uv^2w = a^{N+1+i} b^N a b \notin L$

$\Rightarrow$ L non è regolare!

4) $b^*(a|\varepsilon)b^*$



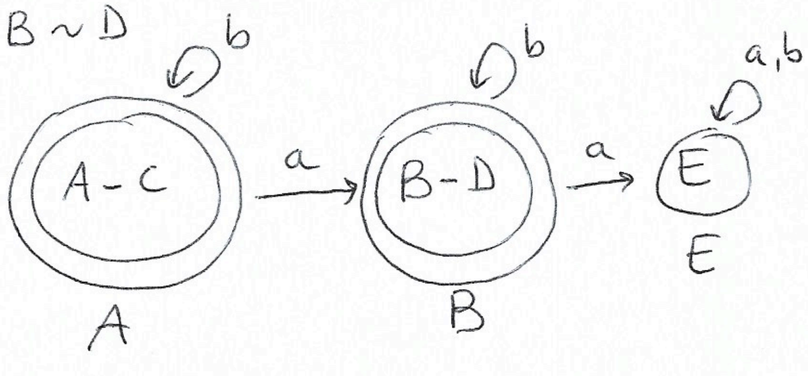


5)

B	X1			
C		X1		
D	X1		X1	
E	X0	X0	X0	X0
	A	B	C	D

$A \sim C$

$B \sim D$



$$\left.\begin{array}{l} A \to bA \mid aB \mid \varepsilon \\ B \to bB \mid \cancel{aE} \mid \varepsilon \\ \cancel{E \to aE \mid bE} \end{array}\right\} \to \begin{array}{l} A \to bA \mid aB \mid \varepsilon \\ B \to bB \mid \varepsilon \end{array}$$

$B \rightsquigarrow b^*$

$A \rightsquigarrow bA \mid ab^* \mid \varepsilon$

$b^*(ab^* \mid \varepsilon)$

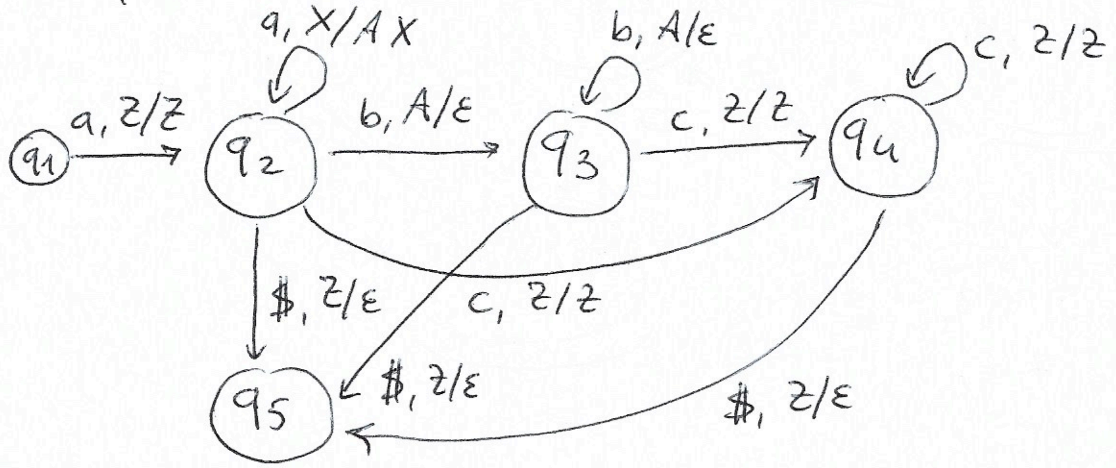
6) L_1 libero L_2 regolare $L_1 - L_2$?

$$L_1 - L_2 = L_1 \cap \overline{L}_2$$

Se L_2 è regolare, allora $\overline{L}_2$ è pure regolare.
L'intersezione tra L_1 libero e $\overline{L}_2$ regolare è libera per un ben noto teorema visto a lezione.

---

7) $L = \{ a^{n+1} b^n c^m \mid n, m \geq 0 \}$ $L\$$ per pila vuota



Non è possibile riconoscere L per pila vuota perché L non gode della prefix property; ad esempio:

$\underline{a} \in L$ ed anche $\underline{ac} \in L$

8) $G \begin{cases} S \to AB \mid BC \\ A \to \varepsilon \mid bD \\ B \to a \mid bCB \\ C \to \varepsilon \mid Cd \\ D \to c \mid dSD \end{cases}$

	First	Follow
S	a, b	\$, c, d
A	ε, b	a, b
B	a, b	\$, c, d
C	ε, d	\$, c, d, a, b
D	c, d	a, b

G non è di classe LL(1) perché, ad esempio, è ricorsiva sx su C. Oppure perché $A \to \varepsilon \mid bD$ è tale che $Follow(A) \cap First(bD) = \{b\}$.

Rimuoviamo la ricorsione sx immediata:

$G' \begin{cases} S \to AB \mid BC \\ A \to \varepsilon \mid bD \\ B \to a \mid bCB \\ C \to C' \\ C' \to dC' \mid \varepsilon \\ D \to c \mid dSD \end{cases}$

Vale che $L(G') = L(G)$.

Ora rimuoviamo le produzioni ε. Calcoliamo i simboli annullabili:

$N(G) = \{A, C', C\}$

$G'' \begin{cases} S \to AB \mid B \mid BC \\ A \to bD \\ B \to a \mid bCB \mid bB \\ C \to C' \\ C' \to dC' \mid d \\ D \to c \mid dSD \end{cases}$

Vale che $L(G'') = L(G')$ perché $\varepsilon \notin L(G')$.

9) $\left.\begin{array}{l} S \to cAS \mid \varepsilon \\ A \to a \mid aA \end{array}\right] G$

(i) $L(A) = a^+$ $\quad L(S) = (c \cdot a^+)^*$

(ii) G non è di classe LL(1) perché $A \to a \mid aA$ e $First(a) \cap First(aA) = \{a\}$.

(iii) $\left.\begin{array}{l} S \to cAS \mid \varepsilon \\ A \to aA' \\ A' \to \varepsilon \mid A \end{array}\right]$ G' è di classe LL(1)

– $First(cAS) \cap First(\varepsilon) = \emptyset$
$First(cAS) \cap Follow(S) = \emptyset$
$\{c\} \cap \{\$\} = \emptyset$

– $First(\varepsilon) \cap First(A) = \emptyset$
$Follow(A') \cap First(A) = \emptyset$
$\{c, \$\} \cap \{a\} = \emptyset$

	First	Follow
S	ε, c	\$
A	a	c, \$
A'	ε, a	c, \$

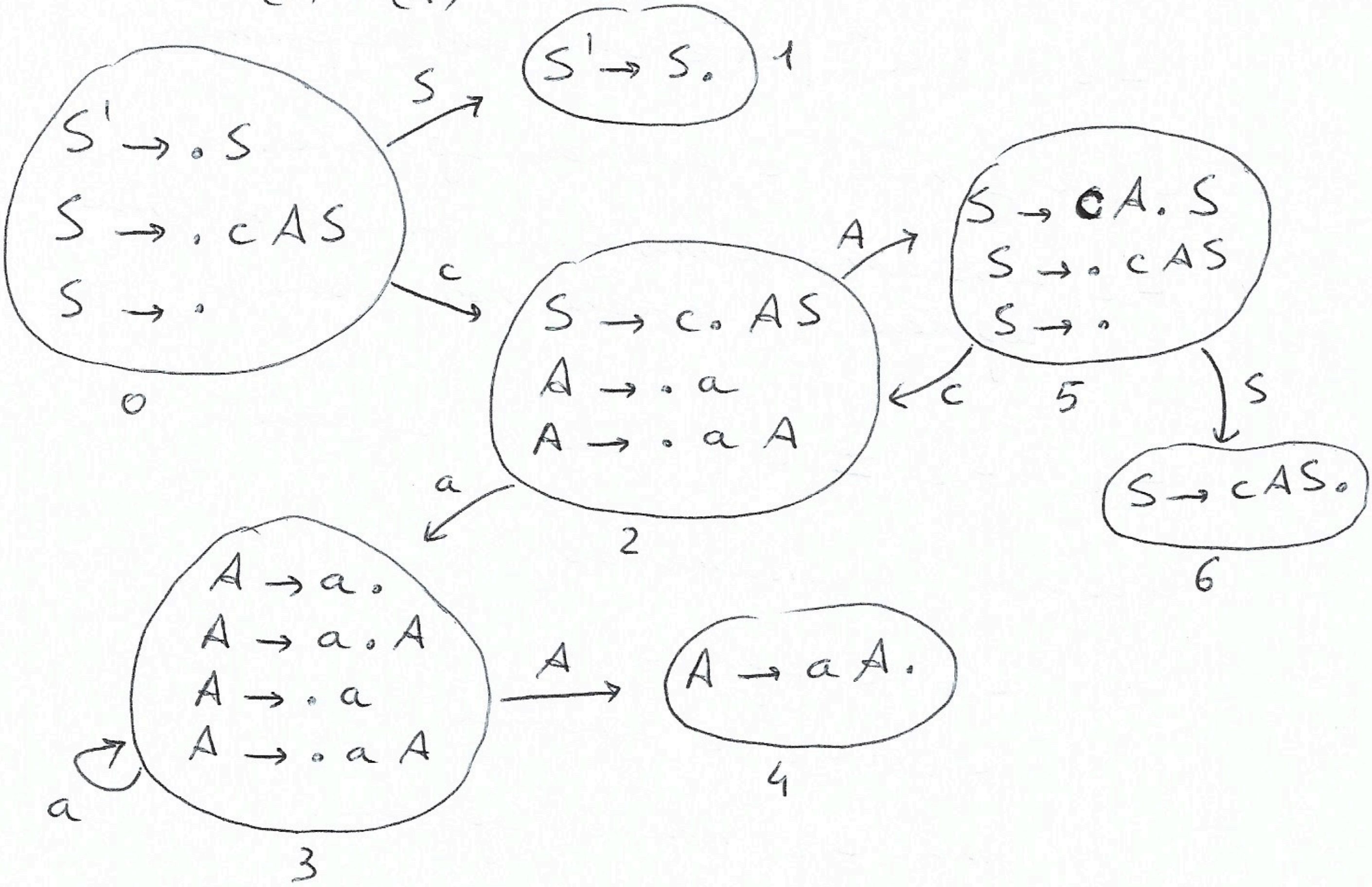
	a	c	\$
S		$S \to cAS$	$S \to \varepsilon$
A	$A \to aA'$		
A'	$A' \to A$	$A' \to \varepsilon$	$A' \to \varepsilon$

Input	Stack
caa\$	S\$
	cAS\$
aa\$	AS\$
	aA'S\$
a\$	A'S\$
	AS\$
	aA'S\$
\$	A'S\$
	S\$
	\$

ok

10) $S \rightarrow \overset{(1)}{cAS} \mid \overset{(2)}{\varepsilon}$

$A \rightarrow \underset{(3)}{a} \mid \underset{(4)}{aA}$



	a	c	$	S	A
0		s2	r2	g1	
1			acc		
2	s3				g5
3	s3	r3	r3		g4
4		r4	r4		
5		s2	r2	g6	
6			r1		

(0, ε, caa $)

(02, c, aa $)

(023, ca, a $)

(0233, caa, $)

$\rightarrow A$

(0234, caA, $)

$\rightarrow A$

(025, cA, $)

$\rightarrow S$

(0256, cAS, $)

$\rightarrow S$

(01, S, $)

<u>Accept!</u>

11) L'affermazione è falsa!

$L_1 = \{aa, aaa\} = \{a^2, a^3\}$

$L_2 = \{a^n \mid n$ è un numero primo$\}$

L_1 è finito e quindi regolare!

L_2: abbiamo dimostrato in classe essere non regolare (e anche non libero!)

$L_1 \wedge L_2 = L_1$ che è regolare.

Quindi, riassumendo:

L_1 è regolare,

$L_1 \wedge L_2$ è regolare,

ma L_2 non è regolare.

Vale invece la proprietà di chiusura:

"se L_1 è regolare ed L_2 è regolare, allora $L_1 \wedge L_2$ è regolare".